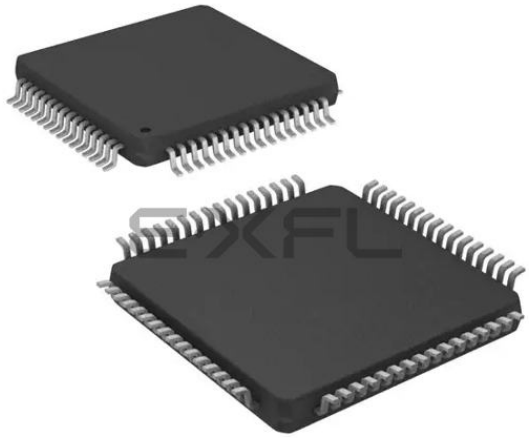
Microchip Technology - PIC32MX564F128H-I/PT Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	MIPS32® M4K™
Core Size	32-Bit Single-Core
Speed	80MHz
Connectivity	CANbus, I²C, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	53
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic32mx564f128h-i-pt

TABLE 2: PIC32MX6XX USB AND ETHERNET FEATURES

USB and Ethernet																
Device	**Pins**	**Program Memory (KB)**	**Data Memory (KB)**	**USB**	**Ethernet**	**Timers/Capture/Compare**	**DMA Channels (Programmable/Dedicated)**	**UART(2,3)**	**SPI(3)**	**I²C(3)**	**10-bit 1 Msps ADC (Channels)**	**Comparators**	**PMP/PSP**	**JTAG**	**Trace**	**Packages(4)**
PIC32MX664F064H	64	64 + 12(1)	32	1	1	5/5/5	4/4	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX664F128H	64	128 + 12(1)	32	1	1	5/5/5	4/4	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX675F256H	64	256 + 12(1)	64	1	1	5/5/5	8/4	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX675F512H	64	512 + 12(1)	64	1	1	5/5/5	8/4	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX695F512H	64	512 + 12(1)	128	1	1	5/5/5	8/4	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX664F064L	100	64 + 12(1)	32	1	1	5/5/5	4/4	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG
PIC32MX664F128L	100	128 + 12(1)	32	1	1	5/5/5	4/4	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG
PIC32MX675F256L	100	256 + 12(1)	64	1	1	5/5/5	8/4	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG
PIC32MX675F512L	100	512 + 12(1)	64	1	1	5/5/5	8/4	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG, TL
PIC32MX695F512L	100	512 + 12(1)	128	1	1	5/5/5	8/4	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG, TL

Legend: PF, PT = TQFP MR = QFN BG = TFBGA TL = VTLA(5)

Note 1: This device features 12 KB boot Flash memory.

2: CTS and RTS pins may not be available for all UART modules. Refer to the **"Device Pin Tables"** section for more information.

3: Some pins between the UART, SPI and I^2C modules may be shared. Refer to the **"Device Pin Tables"** section for more information.

4: Refer to **34.0 "Packaging Information"** for more information.

5: 100-pin devices other than those listed here are available in the VTLA package upon request. Please contact your local Microchip Sales Office for details.

TABLE 1-1: PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number(1)				Pin Type	Buffer Type	Description
	64-Pin QFN/TQFP	100-Pin TQFP	121-Pin TFBGA	124-pin VTLA			
TMS	23	17	G3	B9	I	ST	JTAG Test mode select pin
TCK	27	38	J6	A26	I	ST	JTAG test clock input pin
TDI	28	60	G11	A40	I	ST	JTAG test data input pin
TDO	24	61	G9	B33	O	—	JTAG test data output pin
RTCC	42	68	E9	B37	O	—	Real-Time Clock alarm output
CVREF-	15	28	L2	A21	I	Analog	Comparator Voltage Reference (low)
CVREF+	16	29	K3	B17	I	Analog	Comparator Voltage Reference (high)
CVREFOUT	23	34	L5	A24	O	Analog	Comparator Voltage Reference output
C1IN-	12	21	H2	B11	I	Analog	Comparator 1 negative input
C1IN+	11	20	H1	A12	I	Analog	Comparator 1 positive input
C1OUT	21	32	K4	A23	O	—	Comparator 1 output
C2IN-	14	23	J2	B13	I	Analog	Comparator 2 negative input
C2IN+	13	22	J1	A13	I	Analog	Comparator 2 positive input
C2OUT	22	33	L4	B19	O	—	Comparator 2 output
PMA0	30	44	L8	A29	I/O	TTL/ST	Parallel Master Port Address bit 0 input (Buffered Slave modes) and output (Master modes)
PMA1	29	43	K7	B24	I/O	TTL/ST	Parallel Master Port Address bit 1 input (Buffered Slave modes) and output (Master modes)
PMA2	8	14	F3	A9	O	—	Parallel Master Port address (Demultiplexed Master modes)
PMA3	6	12	F2	A8	O	—	
PMA4	5	11	F4	B6	O	—	
PMA5	4	10	E3	A7	O	—	
PMA6	16	29	K3	B17	O	—	
PMA7	22	28	L2	A21	O	—	
PMA8	32	50	L11	A32	O	—	
PMA9	31	49	L10	B27	O	—	
PMA10	28	42	L7	A28	O	—	
PMA11	27	41	J7	B23	O	—	
PMA12	24	35	J5	B20	O	—	
PMA13	23	34	L5	A24	O	—	
PMA14	45	71	C11	A46	O	—	
PMA15	44	70	D11	B38	O	—	
PMCS1	45	71	C11	A46	O	—	Parallel Master Port Chip Select 1 strobe
PMCS2	44	70	D11	B38	O	—	Parallel Master Port Chip Select 2 strobe

Legend: CMOS = CMOS compatible input or output; Analog = Analog input; P = Power
ST = Schmitt Trigger input with CMOS levels; O = Output; I = Input
TTL = TTL input buffer

Note 1: Pin numbers are only provided for reference. See the **"Device Pin Tables"** section for device pin availability.

2: See **25.0 "Ethernet Controller"** for more information.

3.0 CPU

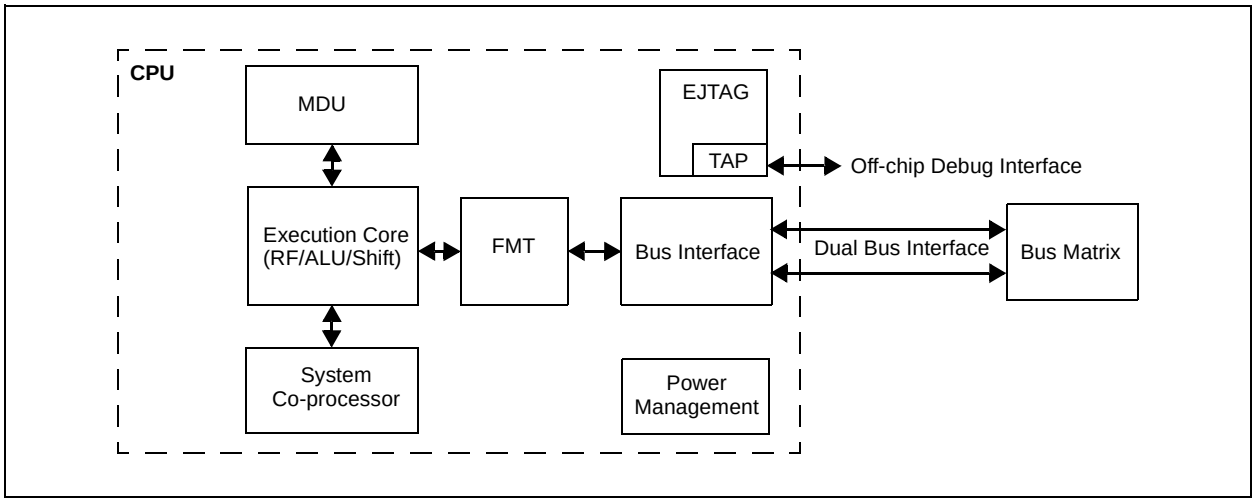
Note: This data sheet summarizes the features of the PIC32MX5XX/6XX/7XX family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 2. "CPU"** (DS60001113) in the *"PIC32 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com/PIC32). Resources for the MIPS32® M4K® Processor Core are available at http://www.imgtec.com.

The MIPS32® M4K® Processor core is the heart of the PIC32MX5XX/6XX/7XX family processor. The CPU fetches instructions, decodes each instruction, fetches source operands, executes each instruction and writes the results of instruction execution to the proper destinations.

3.1 Features

- 5-stage pipeline
- 32-bit address and data paths
- MIPS32 Enhanced Architecture (Release 2)
  - Multiply-accumulate and multiply-subtract instructions
  - Targeted multiply instruction
  - Zero/One detect instructions
  - `WAIT` instruction
  - Conditional move instructions (`MOVN`, `MOVZ`)
  - Vectored interrupts
  - Programmable exception vector base
  - Atomic interrupt enable/disable
  - GPR shadow registers to minimize latency for interrupt handlers
  - Bit field manipulation instructions
- MIPS16e® code compression
  - 16-bit encoding of 32-bit instructions to improve code density
  - Special PC-relative instructions for efficient loading of addresses and constants
  - `SAVE` and `RESTORE` macro instructions for setting up and tearing down stack frames within subroutines
  - Improved support for handling 8-bit and 16-bit data types
- Simple Fixed Mapping Translation (FMT) mechanism
- Simple dual bus interface
  - Independent 32-bit address and data busses
  - Transactions can be aborted to improve interrupt latency
- Autonomous multiply/divide unit
  - Maximum issue rate of one 32x16 multiply per clock
  - Maximum issue rate of one 32x32 multiply every other clock
  - Early-in iterative divide. Minimum 11 and maximum 33 clock latency (dividend (*rs*) sign extension-dependent)
- Power control
  - Minimum frequency: 0 MHz
  - Low-Power mode (triggered by `WAIT` instruction)
  - Extensive use of local gated clocks
- EJTAG debug and instruction trace
  - Support for single stepping
  - Virtual instruction and data address/value
  - Breakpoints
  - PC tracing with trace compression

FIGURE 3-1: MIPS32® M4K® PROCESSOR CORE BLOCK DIAGRAM



NOTES:

7.0 INTERRUPT CONTROLLER

Note: This data sheet summarizes the features of the PIC32MX5XX/6XX/7XX family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 8. "Interrupts"** (DS60001108) in the *"PIC32 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com/PIC32).

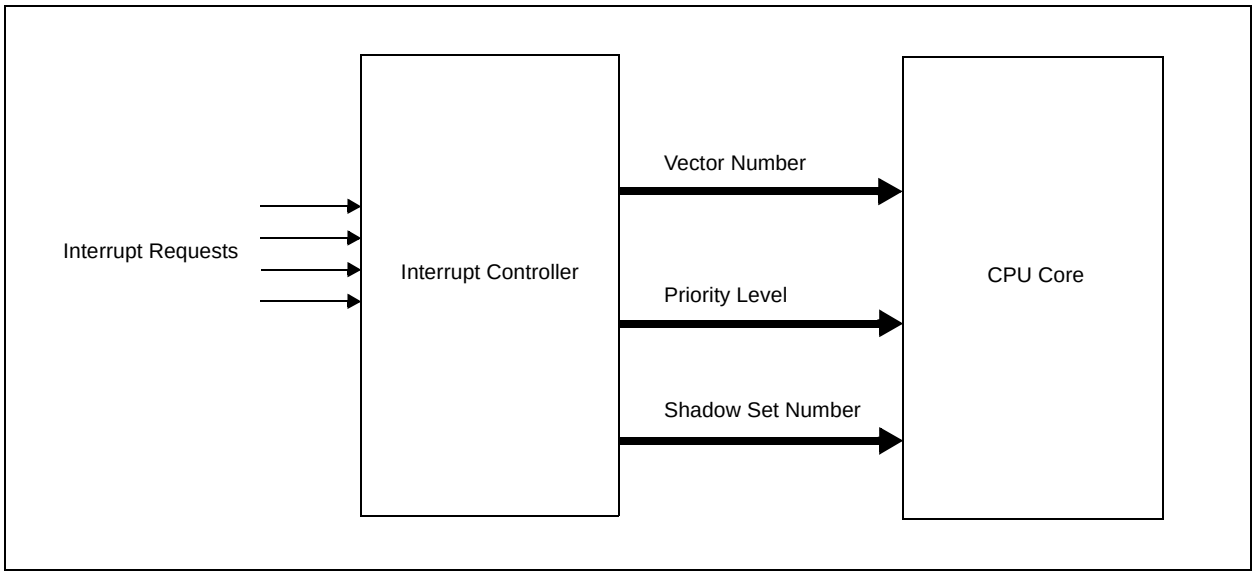
PIC32MX5XX/6XX/7XX devices generate interrupt requests in response to interrupt events from peripheral modules. The interrupt control module exists externally to the CPU logic and prioritizes the interrupt events before presenting them to the CPU.

The Interrupt Controller module includes the following features:

- Up to 96 interrupt sources
- Up to 64 interrupt vectors
- Single and multi-vector mode operations
- Five external interrupts with edge polarity control
- Interrupt proximity timer
- Seven user-selectable priority levels for each vector
- Four user-selectable sub-priority levels within each priority
- Dedicated shadow set for user-selectable priority level
- Software can generate any interrupt
- User-configurable interrupt vector table location
- User-configurable interrupt vector spacing

A simplified block diagram of the Interrupt Controller module is illustrated in Figure 7-1.

FIGURE 7-1: INTERRUPT CONTROLLER MODULE



REGISTER 10-8: DCHxECON: DMA CHANNEL 'x' EVENT CONTROL REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	CHAIRQ<7:0>(1)							
15:8	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	CHSIRQ<7:0>(1)							
7:0	S-0	S-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
	CFORCE	CABORT	PATEN	SIRQEN	AIRQEN	—	—	—

Legend:	S = Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-24 **Unimplemented:** Read as '0'

bit 23-16 **CHAIRQ<7:0>:** Channel Transfer Abort IRQ bits(1)

11111111 = Interrupt 255 will abort any transfers in progress and set CHAIF flag
•
•
•
00000001 = Interrupt 1 will abort any transfers in progress and set CHAIF flag
00000000 = Interrupt 0 will abort any transfers in progress and set CHAIF flag

bit 15-8 **CHSIRQ<7:0>:** Channel Transfer Start IRQ bits(1)

11111111 = Interrupt 255 will initiate a DMA transfer
•
•
•
00000001 = Interrupt 1 will initiate a DMA transfer
00000000 = Interrupt 0 will initiate a DMA transfer

bit 7 **CFORCE:** DMA Forced Transfer bit

1 = A DMA transfer is forced to begin when this bit is written to a '1'
0 = This bit always reads '0'

bit 6 **CABORT:** DMA Abort Transfer bit

1 = A DMA transfer is aborted when this bit is written to a '1'
0 = This bit always reads '0'

bit 5 **PATEN:** Channel Pattern Match Abort Enable bit

1 = Abort transfer and clear CHEN on pattern match
0 = Pattern match is disabled

bit 4 **SIRQEN:** Channel Start IRQ Enable bit

1 = Start channel cell transfer if an interrupt matching CHSIRQ occurs
0 = Interrupt number CHSIRQ is ignored and does not start a transfer

bit 3 **AIRQEN:** Channel Abort IRQ Enable bit

1 = Channel transfer is aborted if an interrupt matching CHAIRQ occurs
0 = Interrupt number CHAIRQ is ignored and does not terminate a transfer

bit 2-0 **Unimplemented:** Read as '0'

Note 1: See **Table 7-1: "Interrupt IRQ, Vector and Bit Location"** for the list of available interrupt IRQ sources.

11.0 USB ON-THE-GO (OTG)

Note: This data sheet summarizes the features of the PIC32MX5XX/6XX/7XX family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 27. "USB On-The-Go (OTG)"** (DS60001126) in the *"PIC32 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com/PIC32).

The Universal Serial Bus (USB) module contains analog and digital components to provide a USB 2.0 full-speed and low-speed embedded Host, full-speed Device or OTG implementation with a minimum of external components. This module in Host mode is intended for use as an embedded host and therefore does not implement a UHCI or OHCI controller.

The USB module consists of the clock generator, the USB voltage comparators, the transceiver, the Serial Interface Engine (SIE), a dedicated USB DMA controller, pull-up and pull-down resistors, and the register interface. A block diagram of the PIC32 USB OTG module is presented in Figure 11-1.

The clock generator provides the 48 MHz clock required for USB full-speed and low-speed communication. The voltage comparators monitor the voltage on the VBUS pin to determine the state of the bus. The transceiver provides the analog translation between the USB bus and the digital logic. The SIE is a state machine that transfers data to and from the endpoint buffers and generates the hardware protocol for data transfers. The USB DMA controller transfers data between the data buffers in RAM and the SIE. The integrated pull-up and pull-down resistors eliminate the need for external signaling components. The register interface allows the CPU to configure and communicate with the module.

The USB module includes the following features:

- USB Full-speed support for host and device
- Low-speed host support
- USB OTG support
- Integrated signaling resistors
- Integrated analog comparators for VBUS monitoring
- Integrated USB transceiver
- Transaction handshaking performed by hardware
- Endpoint buffering anywhere in system RAM
- Integrated DMA to access system RAM and Flash

Note: The implementation and use of the USB specifications, as well as other third party specifications or technologies, may require licensing; including, but not limited to, USB Implementers Forum, Inc. (also referred to as USB-IF). The user is fully responsible for investigating and satisfying any applicable licensing obligations.

15.1 Control Registers

TABLE 15-1: WATCHDOG TIMER REGISTER MAP

Virtual Address (BF80_#)	Register Name(1)	Bit Range	Bits																All Resets(2)
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
0000	WDTCON	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	ON	—	—	—	—	—	—	—	—	SWDTPS<4:0>					—	WDTCLR	0000

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: All registers in this table have corresponding CLR, SET and INV registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, respectively. See **Section 12.1.1 "CLR, SET and INV Registers"** for more information.

2: Reset values are dependent on the DEVCFGx Configuration bits and the type of Reset.

REGISTER 19-2: I2CxSTAT: I^{2}C STATUS REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC
	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
7:0	R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF

Legend:	HS = Set by hardware	HSC = Hardware set/cleared	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	C = Clearable bit

bit 31-16 **Unimplemented:** Read as '0'

bit 15 **ACKSTAT:** Acknowledge Status bit (when operating as I^{2}C master, applicable to master transmit operation)
This bit is set or cleared by hardware at the end of a slave Acknowledge.
1 = NACK received from slave
0 = ACK received from slave

bit 14 **TRSTAT:** Transmit Status bit (when operating as I^{2}C master, applicable to master transmit operation)
This bit is set by hardware at the beginning of a master transmission, and is cleared by hardware at the end of a slave Acknowledge.
1 = Master transmit is in progress (8 bits + ACK)
0 = Master transmit is not in progress

bit 13-11 **Unimplemented:** Read as '0'

bit 10 **BCL:** Master Bus Collision Detect bit
This bit is set by hardware at the detection of a bus collision.
1 = A bus collision has been detected during a master operation
0 = No collision

bit 9 **GCSTAT:** General Call Status bit
This bit is set by hardware when the address matches the general call address, and is cleared by hardware clear at a Stop detection.
1 = General call address was received
0 = General call address was not received

bit 8 **ADD10:** 10-bit Address Status bit
This bit is set by hardware upon a match of the 2nd byte of the matched 10-bit address, and is cleared by hardware at a Stop detection.
1 = 10-bit address was matched
0 = 10-bit address was not matched

bit 7 **IWCOL:** Write Collision Detect bit
This bit is set by hardware at the occurrence of a write to I2CxTRN while busy (cleared by software).
1 = An attempt to write the I2CxTRN register failed because the I^{2}C module is busy
0 = No collision

bit 6 **I2COV:** Receive Overflow Flag bit
This bit is set by hardware at an attempt to transfer I2CxRSR to I2CxRCV (cleared by software).
1 = A byte was received while the I2CxRCV register is still holding the previous byte
0 = No overflow

REGISTER 24-16: CiFLTCON6: CAN FILTER CONTROL REGISTER 6 (CONTINUED)

bit 15 **FLTEN25:** Filter 25 Enable bit
1 = Filter is enabled
0 = Filter is disabled

bit 14-13 **MSEL25<1:0>:** Filter 25 Mask Select bits
11 = Acceptance Mask 3 selected
10 = Acceptance Mask 2 selected
01 = Acceptance Mask 1 selected
00 = Acceptance Mask 0 selected

bit 12-8 **FSEL25<4:0>:** FIFO Selection bits
11111 = Message matching filter is stored in FIFO buffer 31
11110 = Message matching filter is stored in FIFO buffer 30
•
•
•
00001 = Message matching filter is stored in FIFO buffer 1
00000 = Message matching filter is stored in FIFO buffer 0

bit 7 **FLTEN24:** Filter 24 Enable bit
1 = Filter is enabled
0 = Filter is disabled

bit 6-5 **MSEL24<1:0>:** Filter 24 Mask Select bits
11 = Acceptance Mask 3 selected
10 = Acceptance Mask 2 selected
01 = Acceptance Mask 1 selected
00 = Acceptance Mask 0 selected

bit 4-0 **FSEL24<4:0>:** FIFO Selection bits
11111 = Message matching filter is stored in FIFO buffer 31
11110 = Message matching filter is stored in FIFO buffer 30
•
•
•
00001 = Message matching filter is stored in FIFO buffer 1
00000 = Message matching filter is stored in FIFO buffer 0

Note:	The bits in this register can only be modified if the corresponding filter enable (FLTENn) bit is '0'.

REGISTER 24-22: CiFIFOUAn: CAN FIFO USER ADDRESS REGISTER 'n' (n = 0 THROUGH 31)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
	CiFIFOUAn<31:24>							
23:16	R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
	CiFIFOUAn<23:16>							
15:8	R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
	CiFIFOUAn<15:8>							
7:0	R-x	R-x	R-x	R-x	R-x	R-x	R-0(1)	R-0(1)
	CiFIFOUAn<7:0>							

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-0 **CiFIFOUAn<31:0>:** CAN FIFO User Address bits

TXEN = 1: (FIFO configured as a transmit buffer)

A read of this register will return the address where the next message is to be written (FIFO head).

TXEN = 0: (FIFO configured as a receive buffer)

A read of this register will return the address where the next message is to be read (FIFO tail).

Note 1: This bit will always read '0', which forces byte-alignment of messages.

Note: This register is not guaranteed to read correctly in Configuration mode, and should only be accessed when the module is not in Configuration mode.

REGISTER 24-23: CiFIFOCIN: CAN MODULE MESSAGE INDEX REGISTER 'n' (n = 0 THROUGH 31)

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	U-0	U-0	U-0	R-0	R-0	R-0	R-0	R-0
	—	—	—	CiFIFOCI<4:0>				

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-5 **Unimplemented:** Read as '0'

bit 4-0 **CiFIFOCIn<4:0>:** CAN Side FIFO Message Index bits

TXEN = 1: (FIFO configured as a transmit buffer)

A read of this register will return an index to the message that the FIFO will next attempt to transmit.

TXEN = 0: (FIFO configured as a receive buffer)

A read of this register will return an index to the message that the FIFO will use to save the next message.

TABLE 25-5: ETHERNET CONTROLLER REGISTER SUMMARY FOR PIC32MX664F064H, PIC32MX664F128H, PIC32MX664F064L, PIC32MX664F128L, PIC32MX675F256H, PIC32MX675F512H, PIC32MX695F512H, PIC32MX775F256H, PIC32MX775F512H, PIC32MX795F512H, PIC32MX695F512L, PIC32MX675F256L, PIC32MX675F512L, PIC32MX764F128H, PIC32MX764F128L, PIC32MX775F256L, PIC32MX775F512L AND PIC32MX795F512L DEVICES (CONTINUED)

Virtual Address (BF88_#)	Register Name(1)	Bit Range	Bits															All Resets
			31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	
9260	EMAC1 SUPP	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	RESET RMII	—	—	SPEED RMII	—	—	—	—	—	—	—	—	1000
9270	EMAC1 TEST	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	TESTBP	TESTPAUSE	SHRTQNTA	0000
9280	EMAC1 MCFG	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	RESET MGMT	—	—	—	—	—	—	—	—	—	CLKSEL<3:0>				NOPRE	SCANINC	0020
9290	EMAC1 MCMD	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCAN	READ	0000
92A0	EMAC1 MADR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	PHYADDR<4:0>					—	—	—	REGADDR<4:0>					0100
92B0	EMAC1 MWTD	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	MWTD<15:0>																0000
92C0	EMAC1 MRDD	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	MRDD<15:0>																0000
92D0	EMAC1 MIND	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	LINKFAIL	NOTVALID	SCAN	MIIMBUSY	0000
9300	EMAC1 SA0(2)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
		15:0	STNADDR6<7:0>								STNADDR5<7:0>								xxxx
9310	EMAC1 SA1(2)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
		15:0	STNADDR4<7:0>								STNADDR3<7:0>								xxxx
9320	EMAC1 SA2(2)	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
		15:0	STNADDR2<7:0>								STNADDR1<7:0>								xxxx

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: All registers in this table (with the exception of ETHSTAT) have corresponding CLR, SET and INV registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, respectively. See **Section 12.1.1 "CLR, SET and INV Registers"** for more information.

2: Reset values default to the factory programmed value.

REGISTER 25-39: EMAC1SA2: ETHERNET CONTROLLER MAC STATION ADDRESS 2 REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P
	STNADDR2<7:0>							
7:0	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P	R/W-P
	STNADDR1<7:0>							

Legend:		P = Programmable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 31-16 **Reserved:** Maintain as '0'; ignore read

bit 15-8 **STNADDR2<7:0>:** Station Address Octet 2 bits
These bits hold the second transmitted octet of the station address.

bit 7-0 **STNADDR1<7:0>:** Station Address Octet 1 bits
These bits hold the most significant (first transmitted) octet of the station address.

Note 1: Both 16-bit and 32-bit accesses are allowed to these registers (including the SET, CLR and INV registers). 8-bit accesses are not allowed and are ignored by the hardware.

2: This register is loaded at reset from the factory preprogrammed station address.

30.0 INSTRUCTION SET

The PIC32MX5XX/6XX/7XX family instruction set complies with the MIPS32 Release 2 instruction set architecture. The PIC32 device family does not support the following features:

- Core Extend instructions
- Coprocessor 1 instructions
- Coprocessor 2 instructions

Note: Refer to *"MIPS32® Architecture for Programmers Volume II: The MIPS32® Instruction Set"* at www.imgtec.com for more information.

TABLE 32-6: DC CHARACTERISTICS: IDLE CURRENT (IIDLE)

DC CHARACTERISTICS			Standard Operating Conditions: 2.3V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +105°C for V-Temp			
Parameter No.	**Typical(2)**	**Max.**	**Units**	**Conditions**		
Idle Current (IIDLE)(1,3) for PIC32MX575/675/695/775/795 Family Devices						
DC30	4.5	6.5	mA	-40ºC, +25ºC, +85ºC	—	4 MHz
DC30b	5	7		+105°C		
DC31	13	15	mA	-40ºC, +25ºC, +85ºC	—	25 MHz
DC32	28	30	mA	-40ºC, +25ºC, +85ºC	—	60 MHz
DC33	36	42	mA	-40ºC, +25ºC, +85ºC	—	80 MHz
DC33b	39	45	mA	+105°C		
DC34	—	40	µA	-40°C	2.3V	LPRC (31 kHz)
DC34a		75		+25°C		
DC34b		800		+85°C		
DC34c		1000		+105°C		
DC35	35	—	µA	-40°C	3.3V	
DC35a	65			+25°C		
DC35b	600			+85°C		
DC35c	800			+105°C		
DC36	—	43	µA	-40°C	3.6V	
DC36a		106		+25°C		
DC36b		800		+85°C		
DC36c		1000		+105°C		

Note 1: The test conditions for IIDLE current measurements are as follows:

- Oscillator mode is EC (for 8 MHz and below) and EC+PLL (for above 8 MHz) with OSC1 driven by external square wave from rail-to-rail, (OSC1 input clock input over/undershoot < 100 mV required)
- OSC2/CLKO is configured as an I/O input pin
- USB PLL oscillator is disabled if the USB module is implemented, PBCLK divisor = 1:8
- CPU is in Idle mode, program Flash memory Wait states = 111, Program Cache and Prefetch are disabled and SRAM data memory Wait states = 1
- No peripheral modules are operating, (ON bit = 0)
- WDT, Clock Switching, Fail-Safe Clock Monitor, and Secondary Oscillator are disabled
- All I/O pins are configured as inputs and pulled to VSS
- $\overline{MCLR}$ = VDD
- RTCC and JTAG are disabled

2: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

3: This parameter is characterized, but not tested in manufacturing.

4: All parameters are characterized, but only those parameters listed for 4 MHz and 80 MHz are tested at 3.3V in manufacturing.

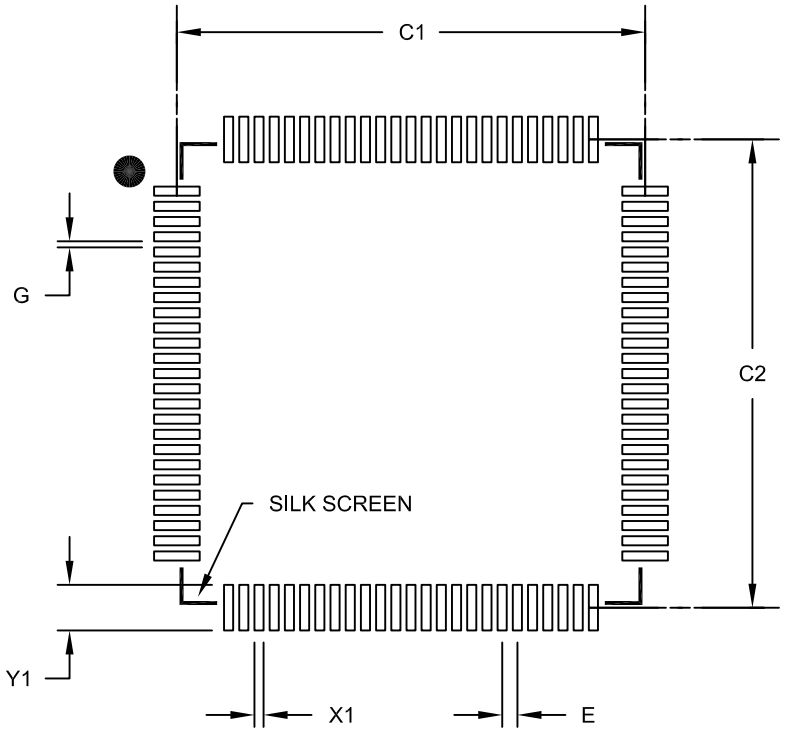
TABLE 32-24: TIMER2, 3, 4, 5 EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 2.3V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +105°C for V-Temp				
Param. No.	**Symbol**	**Characteristics(1)**		**Min.**	**Max.**	**Units**	**Conditions**	
TB10	TTXH	TxCK High Time	Synchronous, with prescaler	[(12.5 ns or 1 TPB)/N] + 25 ns	—	ns	Must also meet parameter TB15	N = prescale value (1, 2, 4, 8, 16, 32, 64, 256)
TB11	TTXL	TxCK Low Time	Synchronous, with prescaler	[(12.5 ns or 1 TPB)/N] + 25 ns	—	ns	Must also meet parameter TB15	
TB15	TTXP	TxCK Input Period	Synchronous, with prescaler	[(Greater of [(25 ns or 2 TPB)/N] + 30 ns	—	ns	VDD > 2.7V	
				[(Greater of [(25 ns or 2 TPB)/N] + 50 ns	—	ns	VDD < 2.7V	
TB20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		—	1	TPB	—	

Note 1: These parameters are characterized, but not tested in manufacturing.

100-Lead Plastic Thin Quad Flatpack (PF) - 14x14x1 mm Body 2.00 mm Footprint [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Contact Pad Spacing	C1		15.40	
Contact Pad Spacing	C2		15.40	
Contact Pad Width (X100)	X1			0.30
Contact Pad Length (X100)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2110B

APPENDIX A: MIGRATING FROM PIC32MX3XX/4XX TO PIC32MX5XX/6XX/7XX DEVICES

This appendix provides an overview of considerations for migrating from PIC32MX3XX/4XX devices to the PIC32MX5XX/6XX/7XX family of devices. The code developed for the PIC32MX3XX/4XX devices can be ported to the PIC32MX5XX/6XX/7XX devices after making the appropriate changes outlined below.

A.1 DMA

PIC32MX5XX/6XX/7XX devices do not support stopping DMA transfers in Idle mode.

A.2 Interrupts

PIC32MX5XX/6XX/7XX devices have persistent interrupts for some of the peripheral modules. This means that the interrupt condition for these peripherals must be cleared before the interrupt flag can be cleared.

For example, to clear a UART receive interrupt, the user application must first read the UART Receive register to clear the interrupt condition and then clear the associated UxIF flag to clear the pending UART interrupt. In other words, the UxIF flag cannot be cleared by software until the UART Receive register is read.

Table A-1 outlines the peripherals and associated interrupts that are implemented differently on PIC32MX5XX/6XX/7XX versus PIC32MX3XX/4XX devices.

In addition, on the SPI module, the IRQ numbers for the receive done interrupts were changed from 25 to 24 and the transfer done interrupts were changed from 24 to 25.

TABLE A-1: PIC32MX3XX/4XX VERSUS PIC32MX5XX/6XX/7XX INTERRUPT IMPLEMENTATION DIFFERENCES

Module	Interrupt Implementation
Input Capture	To clear an interrupt source, read the Buffer Result (ICxBUF) register to obtain the number of capture results in the buffer that are below the interrupt threshold (specified by ICI<1:0> bits).
SPI	Receive and transmit interrupts are controlled by the SRXISEL<1:0> and STXISEL<1:0> bits, respectively. To clear an interrupt source, data must be written to, or read from, the SPIxBUF register to obtain the number of data to receive/transmit below the level specified by the SRXISEL<1:0> and STXISEL<1:0> bits.
UART	TX interrupt will be generated as soon as the UART module is enabled. Receive and transmit interrupts are controlled by the URXISEL<1:0> and UTXISEL<1:0> bits, respectively. To clear an interrupt source, data must be read from, or written to, the UxRXREG or UxTXREG registers to obtain the number of data to receive/transmit below the level specified by the URXISEL<1:0> and UTXISEL<1:0> bits.
ADC	All samples must be read from the result registers (ADC1BUFx) to clear the interrupt source.
PMP	To clear an interrupt source, read the Parallel Master Port Data Input/Output (PMDIN/PMDOUT) register.

INDEX

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

ISBN: 978-1-5224-0958-8